METAPHORS FOR INTERFACE DESIGN

Edwin Hutchins

April 1987

ICS Report 8703

COGNITIVE SCIENCE

UCSD

INSTITUTE FOR COGNITIVE SCIENCE

UNIVERSITY OF CALIFORNIA, SAN DIEGO LA JOLLA, CALIFORNIA 92093

(12)

METAPHORS FOR INTERFACE DESIGN

Edwin Hutchins

April 1987

ICS Report 8703

Institute for Cognitive Science
University of California, San Diego
La Jolla, California 92093

Paper presented at NATO-sponsored workshop on Multimodal Dialogues Including Voice, Venaco, Corsica, France, September 1986.

The research reported here was conducted under Contract N00014-85-C-0133, NR 667-541 with the Personnel Training Research Programs of the Office of Naval Research and with the support of the Navy Personnel Research and Development Center. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsoring agencies. Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government. Requests for reprints should be sent to Edwin Hutchins, Institute for Cognitive Science, C-015; University of California, San Diego; La Jolla, CA 92093.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
ICS 8703

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Institute for Cognitive Science
University of California, San Diego

6b. OFFICE SYMBOL *(If applicable)*

6c. ADDRESS *(City, State, and ZIP Code)*
C-015
La Jolla, CA 92093

7a. NAME OF MONITORING ORGANIZATION
Cognitive Science
Office of Naval Research (Code 1142CS)

7b. ADDRESS *(City, State, and ZIP Code)*
800 North Quincy Street
Arlington, VA 22217-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL *(If applicable)*

8c. ADDRESS *(City, State, and ZIP Code)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-85-C-0133

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
61153N	RR04206	RR04206-0A	NR 667-541

11. TITLE *(Include Security Classification)*
Metaphors for Interface Design

12. PERSONAL AUTHOR(S)
Edwin Hutchins

13a. TYPE OF REPORT
Technical

13b. TIME COVERED
FROM ________ TO 86 Sep

14. DATE OF REPORT *(Year, Month, Day)*
1986 April

15. PAGE COUNT
18

16. SUPPLEMENTARY NOTATION
Paper presented at NATO-sponsored Workshop on Multimodal Dialogues Including Voice, Venaco, Corsica, France, September 1986.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	08	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
human-machine interfaces; speech act theory, metaphor and thought

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Computer system designers and computer users frequently utilize metaphors as organizing structures for dealing with the complexity of behavior of human/computer interfaces. This paper considers four metaphors concerning the mode of interaction between user and machine: the conversation metaphor, the declaration metaphor, the model world metaphor and the collaborative manipulation metaphor. It is argued that the key to the functional properties of an interface lie in the reference relations between the expressions in the interface language and the things to which the expressions refer. The ways in which such metaphors are suggested by advances in I/O technology and the ways they constrain the possibilities we see in technology are discussed. Each of the metaphors discussed promotes a particular type of reference relation. Furthermore, because the computer is a medium in which types of reference relations that are not possible in ordinary language can be realized, the space of interface metaphors is quite likely much larger than we presently imagine it to be.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Dr. Michael G. Shafto

22b. TELEPHONE *(Include Area Code)*
(202) 696-4596

22c. OFFICE SYMBOL
ONR 1142CS

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Metaphors for Interface Design

EDWIN HUTCHINS

Computers are the most plastic medium ever invented for the representation and propagation of information. In fact, they are so adaptable and can manifest such a wide range of behaviors, that little but the hardware itself may be easily identifiable as an enduring property of the device. Computers can mimic the behaviors of other information media and can manifest behaviors that are simply not possible in any other medium. We might speak literally about the nature of the computer's behavior (to the extent we can speak literally about anything) at a very low level, describing the changes in the states of silicon gates and so on, but even there we frequently resort to metaphors. As the levels of complexity are layered one atop the other to produce the high-level behaviors that are the actions we recognize while interacting with the computer, the possibility of talking or thinking literally about the computer's behavior vanishes. We deal with this complexity and this plasticity by speaking metaphorically about the behavior of the computer. The metaphors we use both intentionally and unintentionally, contribute structure in terms of which we organize our understandings of what is going on (Lakoff & Johnson, 1980). My machine, for example, "reads, writes, copies, and edits" files, "flushes" buffers, "creates, refreshes, kills, and buries" windows, "arrests" processes, "inspects, describes, and sends messages to" objects, "calls and traces" functions, and a great deal more. I would have little hope of understanding what the machine can do if I did not have a sense of what sorts of "things" exist in my machine and what sorts of activities those things engage in. This sense is provided, in large part, by an extensive set of metaphors.

TYPES OF INTERFACE METAPHOR

Metaphors are applied to virtually all levels of system behavior. System designers use metaphors when thinking about their designs, and in this way, metaphors may shape the design process. The metaphors also provide a language within the design community that designers use to communicate their designs to each other. Some, like "reading" and "writing" are thoroughly entrenched in the culture of computer design. Metaphors reach the user community as ways of talking about the behavior of the system and here they provide the users with resources for thinking about what the machine is doing. The importance of metaphors in the presentation of computer systems is revealed by the rate at which metaphors are being registered as trademarks in the current highly competitive computer marketplace. Of course, users do not necessarily understand a system the way it is understood by designers and marketing analysts. Users must invent their own interpretations of the metaphors and discover the limits of the mapping of the metaphor onto the behavior of the system. Users sometimes even invent their own metaphors as a means of coming to terms with the behavior of a system. Metaphors are, therefore, not fundamental properties of the system behavior per se. They are, instead, ways of understanding the system's behavior. However, as a convenience, I shall use the names of particular metaphors to refer to interfaces that were designed in accordance with or are well conceived in terms of that metaphor.

There are at least three distinguishable types of metaphor describing various aspects of human-computer interface design.

- *Activity metaphors*. These refer to the user's highest level goals or to the institutional goals that are held for the user whether the user shares them or not. Activity metaphors structure expectations or intentions with respect to the outcome of the interaction. Is the user playing a game? Designing an artifact? Communicating with other humans? Controlling a process?

- *Mode of interaction metaphors*. The reference to "dialogue" in the title of this workshop and many of its papers is an example of the use of a mode of interaction metaphor. These metaphors organize understandings about the nature of the interaction with the computer. Mode of interaction metaphors concern the relationship between the user and the computer without regard for the particular task the user is attempting to accomplish via the computer. The choice of metaphor at this level determines what sort of thing the user thinks the computer is. Is it a conversational partner? An environment for action? A tool box and materials shed?

- *Task domain metaphors*. Task domain metaphors provide the user with a structure for understanding the nature of particular tasks as presented by the computer. A common metaphor for the management of information stored in computers, for example, is the "file" system metaphor. The user can behave as if information is stored in files that have properties something like those of paper files stored in a file cabinet. The computer provides a set of file manipulation operations that may have analogues in the operations one performs on paper files. Material can be added to or deleted from the files, new files can be created, files can be removed from the file system, and so on. Editors, mail programs, terminal emulators, debuggers, and other application packages are built on task domain metaphors that give coherence to the activities they support. Each defines the objects and the operations that exist in the task domain, and each hopefully provides a structure that is easily mappable onto the behaviors of the system.

There is some independence between these types of metaphor. The operations on files provided under the file manipulation metaphor could be invoked under any of several mode of interaction metaphors. The user might specify an action to be taken on a file, for example, by describing the action conversationally, by manipulating controls that cause the action to happen, by issuing a command to execute the action, or by performing in some other mode of interaction. There are also constraints among these types of metaphor. Some mode of interaction metaphors, for example, can only be maintained via the creation of appropriate domain metaphors.

In this paper I am most concerned with metaphors for mode of interaction. Primary attention will be focused upon these four: (a) conversation, (b) declaration, (c) model-world, and (d) collaborative manipulation. I will show how mode of interaction metaphors are essential to the user's interpretation of the behavior of the interface, how interface designers, sometimes unknowingly, encourage particular metaphorical interpretations of the interfaces they design, and how the choice of metaphor has important, but often overlooked, consequences for both the designers and the users of interfaces.

THE CONVERSATION METAPHOR

The metaphor of user and computer engaged in a conversation with each other or carrying on a dialogue about the task at hand is the most popular of the mode of interaction metaphors for human-computer interfaces. This metaphor seems to be based upon a structure of assumptions that goes something like this:

1. The problem of human-computer interfaces is a communication problem. In order to work with each other, the user and the computer must communicate.

2. Human-to-human communication is carried out primarily by means of conversation.

3. Because humans already have considerable skills for interacting with each other, making a computer interface behave like a human permits the human user to utilize already acquired skills, and that makes the interaction easier for the user. That is, human-computer interfaces become more usable the more they mimic human-human interactions.

4. Therefore, human-computer interfaces should support conversation between user and computer.[1]

Consider some of the properties of the conversation metaphor. The conversation metaphor inserts an implied intermediary between the user and the world in which actions are taken (see Figure 1). In a system built on the conversation metaphor, the interface is a language medium in which the user and the system have a conversation about some world. The interface is an implied intermediary between the user and the world about which things are said. In many cases, the world about which things are said is not explicitly represented. In such a setting, the burden is on the user to maintain a model of the state of this unrepresented world. This can be a considerable burden and can lead to many sorts of errors, especially the attempt to carry out actions in inappropriate environments. Alternatively, it can lead to the user making frequent requests to the intermediary to describe or report on the relevant aspects of the task environment, e.g., requesting a listing of file names prior to describing a file system operation. On the other hand, an interface built on the conversational metaphor can take full advantage of the power of abstraction available in symbolic reference. The implied intermediary can be charged with the responsibility of mapping user input expressions onto the world of interest, enabling very economical descriptions. The popularity of the conversation metaphor may be due both to the surface

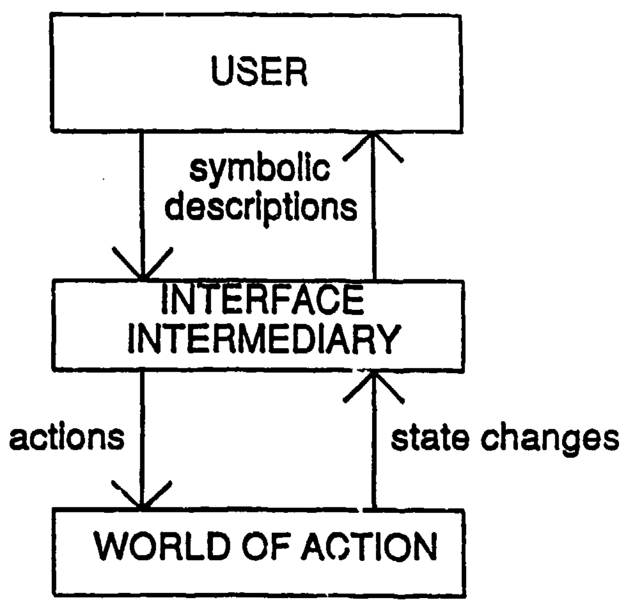

FIGURE 1. The Conversation Interface. Here the user has a conversation with an intermediary who acts on the world of action. The conversation consists of exchanges of symbolic descriptions between user and interface intermediary.

[1] There are good reasons to question each of these assumptions. I will present these reasons at the close of the paper.

credibility of the assumptions on which it is based and to the strength of the teletype legacy. For the first three decades of computer use, the teletype and its technological relatives have been the primary form of interface hardware. Dealing as they do in characters and lines of text, they naturally support, if not a conversation, then at least an exchange of character strings between user and computer. Perhaps we can take batch processing to be the prototype of early human-computer conversation with the participants taking very long conversational turns via card reader and line printer. The teletype permitted shorter conversational turns, but it was still interaction based on the conversation metaphor and it is still very low bandwidth communication. In order to get much done through low bandwidth communication, one needs dense symbols in the interface language; symbols that stand for complicated procedures, for example. This narrowness of bandwidth encourages even more the conception of the computer as an agent that can interpret simple symbols that refer to complicated procedures. Furthermore, this metaphor feeds and is fed by other related metaphors. I do not know which came first historically, the concept of the computer as a brain, the heart of artificial intelligence, or the notion of conversing with it. Clearly, each suggests the other, and as either gains strength so does the other.

Finally, regardless of our metaphorical preferences with regard to mode of interaction, the fact is that every interface implements an interface language in which the user composes expressions that are subsequently interpreted by the computer and in which the computer composes expressions that inform the user of what has happened. That seems like the literal makings of a conversation no matter what we may think.

All of these factors suggest a conversational conception of human-computer interaction. Yet, the conversational metaphor does not quite fit the reality of most human-computer interactions. Typical conversations on "conversational" interfaces are very stilted in a variety of ways discussed by other papers in this workshop. For example, the typical human-machine conversation is conducted with a limited partner via a low bandwidth channel using a severely constrained vocabulary and language syntax. The conversing parties do not mutually repair each other's production errors, and of course, the user's conversational turn typically consists of typing rather than speaking, while the machine's turn consists of displaying characters on a screen. These discrepancies between the metaphorical ideal of human-human conversation and the reality of human-computer conversation form a sort of design vacuum. Having decided upon the desirability of the conversational metaphor, that metaphor now pulls interface technology toward the full realization of the metaphorical potential. If one consults the proceedings of almost any interface design conference, one will find a host of efforts to fill this design vacuum. If only we could use natural language and could speak our input. If only the machine could understand what we mean and talk back to us. Then we would have a truly conversational interface. This is a healthy role for a metaphor, but not one that is usually considered when the metaphor is suggested.

BEYOND CONVERSATION

Recently, something different has been happening in interface design. With the widespread availability of new interface hardware including high-resolution bitmapped displays, pointing devices, and faster processors, a new class of interface has emerged. Literally hundreds of such systems are now available and they appear to be very popular, especially with casual users. It is certainly possible to regard these interfaces using the conversational metaphor. I take references to "visual dialogues," "gestural dialogues," "graphical languages," etc., to be examples of the application of the conversational metaphor to these systems. Schneiderman (1982, 1983) coined the term "direct manipulation" to refer to these systems. The technology on which these systems are based has actually been around for more than 20 years (Sutherland, 1963), but it has only become widely available in the past few years.

The research group with which I am affiliated has been in the business of building interfaces of this type for many years. Examples include a simulation-based steam propulsion training system, Steamer, (Hollan, Hutchins, & Weitzman, 1984), a graphics editor (Hollan, Hutchins, McCandless, Rosenstein, &

Weitzman, in press), a radar navigation training system, and a "direct manipulation" statistical analysis facility (Owen, 1986). Until recently, however, we have not thought very seriously about why these interfaces work the way they do. We believe that an understanding of the cognitive principles that underlie their apparent usability will enable us to build even better interfaces.

Some researchers have tried to identify "direct manipulation" with a particular set of interface behaviors. Schneiderman, for example, uses direct manipulation to refer to systems having the following characteristics:

1. Continuous representation of the objects of interest.
2. Physical actions or labeled button presses instead of complex syntax.
3. Rapid incremental reversible operations whose impact on the object of interest is immediately visible. (1982, p.251)

We believe that a checklist is a weak approach to understanding these interfaces. Even if these are the right characteristics, we would like to know why they are good.

In an earlier paper (Hutchins, Hollan, & Norman, 1985), we described two aspects of the interface that seemed to produce the sensation of directness of action: distance and engagement.

> [Distance] involves a relationship between the task the user has in mind and the way that task can be acccomplished via the interface. Here the critical issues involve minimizing the effort required to bridge the gulf between the user's goals and the way they must be specified to the system. (Hutchins, Hollan, & Norman, 1985, p. 318)

We identified two components of distance in this gulf: *semantic distance* and what I will call here *referential distance*.[2] Figure 2 shows the gulf. Figure 3 shows the relationship between these types of distance.

Semantic distance concerns the relationship between the user's itentions and the meanings of the expressions that are possible in the interface language. It refers to the extent to which the interface language provides means of expressing the user's intentions. Is there a simple expression for what one intends, or is one obliged to construct a lengthy circumlocution? High-level programming languages can be seen as attempts to reduce semantic distance by providing the user with simple expressions (e.g., function names) that refer to frequently encountered problem decompositions.

Referential distance refers to the extent that the user's understanding of the meaning of the expression is similar to the user's understanding of the form of the expression. Symbolic interfaces, for example, are typically high in referential distance because the relationships between the forms of the

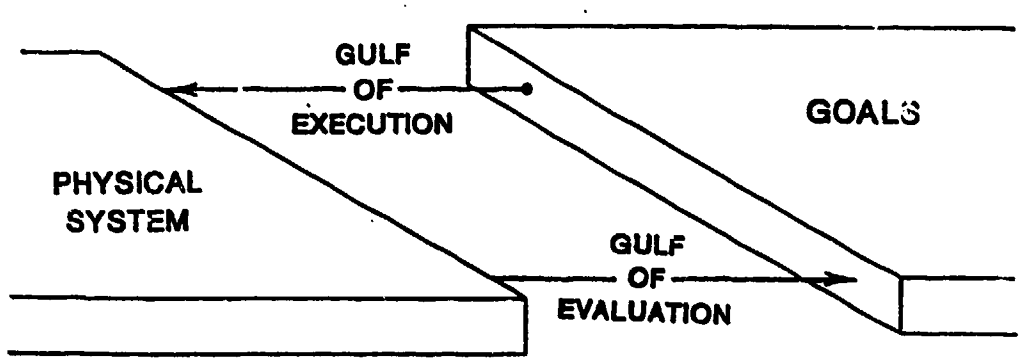

FIGURE 2. The Gulfs of Execution and Evaluation. Each Gulf is unidirectional: The Gulf of Execution extends from user goals to system state; the Gulf of Evaluation extends from system state to user goals.

[2] In the earlier work of Hutchins, Hollan and Norman, this concept was called *articulatory distance*. This name is unfortunate for reasons that should become clear as I explicate the meaning of *referential distance*.

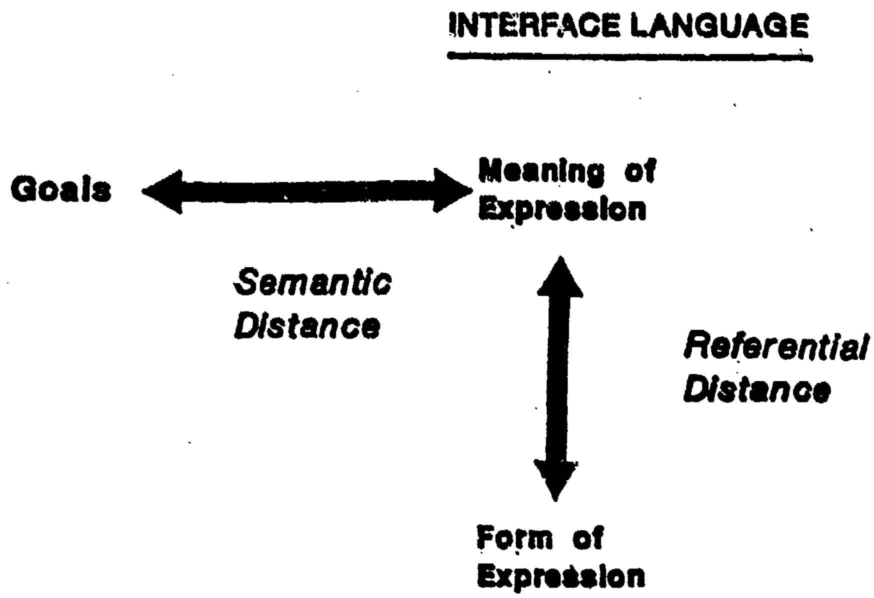

FIGURE 3. Every expression in the interface language has a meaning and a form. Semantic distance reflects the relationship between the user's intentions and the meanings of expressions in the interface language for both input and output. Referential distance reflects the relationship between the physical form of the expression and its meaning. The easier it is to get from the form of the expression to meaning, the smaller the referential distance.

expressions and their meanings are arbitrary. We proposed a cognitive basis for this sensation, arguing that the better the interface to a system helps bridge the gulf between user intention and action, the less cognitive effort needed and the more direct the resulting feeling of interaction.

Engagement proved more difficult to deal with. We felt that

> The systems that best exemplify direct manipulation all give the qualitative feeling that one is directly engaged with the control of objects--not with the programs, not with the computer, but with the semantic objects of our goals and intentions. (Hutchins, Hollan, & Norman, 1985, p. 318)

When it came to specifying how this sensation was to be produced, however, we also resorted to a checklist, not unlike the one proposed by Schneiderman. We did add the condition that the interface language should present to the user a *model world* such that the objects of that world appear and behave as though they are the objects of interest. We knew that the model world was important, but we were stuck thinking about the properties of the interface language. In particular we were implicitly committed to the idea that expressions in the interface had "meanings" that were to be *interpreted* by the machine, in the case of user input expressions, or that were in some sense *intended* by the machine, in the case of machine output. As a consequence, our discussion at that time focused on techniques for reducing referential distance by using expressions that have nonarbitrary relations to their referents. We considered onomatopoea, iconic representation, and located the power of pointing devices in the fact that they are "spatio-mimetic." With the exception of the "spatio-mimetic" nature of pointing devices, these ideas are grounded in the conversational metaphor, and it is not possible to understand the power of the model-world metaphor without shaking them off. At that time we failed to see that while as observers and actors we may certainly intend what we do in the world and interpret the consequences, the world itself neither interprets our actions nor intends the consequences. Our actions happen in the world, but they do not have "meanings" that are interpreted by the world in order to determine how the

world is affected.[3] The notion of "the meaning of an expression" implies a reference gap; a relationship between one thing and some other thing that it "represents" or "stands for." The reference gap in turn implies an interpreter, an agent that can bridge the gap and make a mapping from symbolic expression to referent. This reference gap does not exist for actions in the world, yet it is a fundamental property of symbolic relations, and the power of computers can be traced to their abilities as symbol systems.

I want to argue here that the reasons for the apparent usability of this new class of interfaces lie in the nature of the relationships between expressions in the interface language and the things to which the expressions refer. The key to the sensation of directness in these new interfaces is that these new interface technologies permit the design of an interface under the model-world metaphor. By simulating a world of action, this metaphor collapses the symbolic reference gap. This metaphor does not simply reduce referential distance, it eliminates it! Before we can see how the model-world metaphor does what it does, however, we need to consider the nature and implications of reference relations more broadly.

THE DECLARATION METAPHOR

In the opening essay of *Expression and Meaning*, Searle argues that

> [T]here are a rather limited number of basic things we do with language: we tell people how things are, we try to get them to do things, we commit ourselves to doing things, we express our feelings and attitudes, and we bring about changes through our utterances. (1979, p. 29)

The first four kinds of things we do with language, assertives, directives, commissives, and expressives, respectively, are done with descriptions of the world, but the last thing on Searle's list, bringing about changes through utterances, is different. Searle has termed utterances that do this *declarations*. These are "cases where one brings a state of affairs into existence by declaring it to exist, cases where, so to speak, 'saying makes it so'"(1979, p. 16). Searle gives as examples "I resign," "You're fired," "I excommunicate you," "I appoint you chairman," and others. Successful performance of a declaration guarantees that the propositional content of the utterance corresponds to the world. Searle says, "Declarations bring about some alteration in the status or condition of the referred to object or objects solely in virtue of the fact that the declaration has been successfully performed" (1979, p. 17). What makes these utterances special is their relation to the world to which they refer. Notice that all the objects referred to in the declarations are culturally constructed objects (D'Andrade, 1981). Employment, membership in a church, and the chair of a meeting are all social entities. Each is embedded in a social arrangement in which it is people's agreement that it is so that makes it so. They refer to aspects of the social world that exist only by virtue of the participants agreeing that they exist. The agreements are made and unmade by language acts. These declarations change the world they refer to by changing the agreement under which something does or does not exist. The relation between the expression and the thing to which it refers can therefore be causal rather than simply descriptive. It is the properties of that world that make that causality possible. Declarations are not always successfully performed, but when they are, they have their effects because they refer to a world that can be constructed and modified by the performance of expressions in the language.

The existence in natural language of declarations as a class of speech acts with this special reference relation suggests that the same reference relation could also be supported by computer interfaces that appear to be based on the conversational metaphor. And in fact, some experienced users of such interfaces appear to discover this fact on their own. Consider what it would take to turn a "command language" interface into a "declaration language" interface. The difference between a "command

[3] Of course, actions may have symbolic meanings, but these are meanings that are interpreted by other symbol processing devices, i.e., people, not by the physical world in which they are enacted and in which they may have physical consequences.

language" and a "declaration language" interface is largely in the mind of the user. If the user parses "delete foo" onto the deep structure corresponding to the imperative form "(you) delete foo (from the system)" or "(I command you to) delete foo," then it is a command language interface with the implicit imperative "you" as the implied intermediary. If the user parses "delete foo" as the declaration "(I hereby declare) foo deleted," or even "(I hereby declare) foo deleted (from you, the system)," then it is a declaration interface with no implied intermediary. Of course, the user might still have to keep track of the state of the world acted upon, since it might not be explicitly represented, but this would nevertheless be a declarative interface. Figure 4 shows the relation of user to world of action under the declaration metaphor.

Users could make this shift on their own. Some occasionally seem to do so. Consider a case involving the use of the screen editor in the UNIX operating environment, *vi*. The command *dw*, shorthand for "delete word," is a frequently invoked command in *vi*. Experienced users who have overlearned the command cease regarding it as a instruction to an agent to carry out on the text file and instead regard it as a symbolic incantation that causes the word to the right of the cursor to disappear. In shifting to the declaration metaphor, these users have eliminated the intermediary between themselves and the world of interest.

If declaration became the dominant metaphor for an interface, the user could become a magician for whom every expression in the input language would be a incantation having the power of a declaration. The magician would make the world as it is by declaring it to be so! Of course, the power of such a magician would not lie entirely in either the magician or the language. The power of declarations lies as much in the nature of the world that is referred to as in the utterances that do the referring. Just as declarations in natural language depend upon the culturally constructed nature of the world to which they refer, the declarations in a computer interface language depend upon the special nature of the world of the computer system. And, of course, one of the great virtues of the plasticity of the computer as a medium is that it happens to be a world in which saying something *can* make it so. This is an important, but often overlooked, difference between most uses of natural language and computer interface languages. It is a difference that can be exploited in the design of computer interfaces by establishing reference relations between the interface language and the world to which it refers that permit the user to think of the interface as a magical world.

Both the conversation and the declaration metaphors are implicit options for the user with respect to most so-called conversational interfaces, but few of those who are known for their computational wizardry see themselves as magicians of the declarative sort working directly on the world rather than via an intermediary. I believe there are two major reasons for this. First, there is a strong historical and cultural bias in favor of the conversation metaphor and against the declaration parsing. Considering the

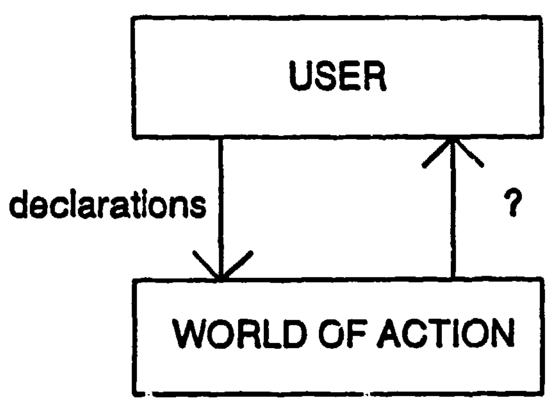

FIGURE 4. The Declaration Interface. Here the user performs declarations, descriptions with causal force, directly in the world of action. What the user observes is not clear. If things go well, state changes are observed, but if the declaration cannot be satisfied by the world, an error message may result. Such an error message destroys the declaration metaphor.

user input side of the interaction, the conversation parsing is implicitly suggested by the name "command language," where commands are issued to agents, and by the popularity of the "conversational" metaphor for interface design in general. More important, perhaps, interfaces that have been designed under the conversational metaphor frequently behave in ways that are difficult to accomodate in the declaration metaphor. For example, many interfaces take advantage of the implied intermediary as a place to locate error messages. Consider an error message stating "delete: Command not found" or "ls: /hutchins/foo: Permission denied." These messages are easy to interpret as advice from an intermediary who has attempted to carry out the command but has been unable to do so, but they are difficult to interpret in the declarative model. If the user is a magician uttering incantations with causal force in the world, who is saying these things? Who couldn't find my command? Who denies permission? These are aspects of the system's behavior that are not captured by any aspect of the declaration metaphor, and in general, the declaration metaphor is difficult to maintain with respect to any interface that produces error messages.

If we were to design an interface with complete fidelity to the declaration metaphor, then when a user generated an unfelicitous, semantically anomalous, or ungrammatical expression, nothing would happen. After all, nothing happens when a magician utters a meaningless or ineffective spell. So the world should not change and there should be no notification of a problem. That reading of the declaration metaphor would surely lead to the design of very frustrating interfaces. A better solution would be to prevent the magician/user from ever uttering such a spell. But how can that be done without invoking an intermediary to monitor and filter the user's utterances?

As it stands, declarations have the power to directly change the world, but nothing rules out impossible declarations. If saying is to be doing, then there must be some way of ensuring that nothing can be said that cannot be done. Otherwise, some intermediary will have to intervene, and that destroys the declaration metaphor in which the magician does by saying. The declaration metaphor, in which "saying is doing," can only be supported if everything that can be said can be done. Giving the declarations direct causal force in the world is half the solution to the problem of supporting a metaphor for more direct action. Constraining the production of declarations is the other half. The trouble with declarations, however, is that they are linguistic entities. They are inherently symbolic, and they exist independently of the things they describe. It is difficult to imagine a natural way to constrain the production of declarations such that only those things that are possible in the world of action can be described. The constraints would surely appear arbitrary because they belong to the domain of action, not to the world of description building. Still, arbitrary or not, these constraints are sometimes embodied in the interface, as, for example, in the use of dynamic menus that only present options that are meaningful in the current task environment.

The declaration metaphor is a metaphor that half works. It is not quite viable, because it inevitably presents the user either with opportunities to enter situations that destroy the metaphor itself or with what seem to be arbitrary constraints on the generation of declarations to be enacted upon the world.

THE MODEL-WORLD METAPHOR

The model-world metaphor can become supportable at virtually all levels of interaction in interfaces utilizing currently available I/O technologies. The two requirements for the maintenance of a model-world metaphor are that expressions in the interface language appear as actions with causal force in the world of interest and that the generation of expressions is constrained such that it is not possible to compose an expression that cannot be realized in the world of interest. Figure 5 shows the relation of user to world of action under the model-world metaphor.

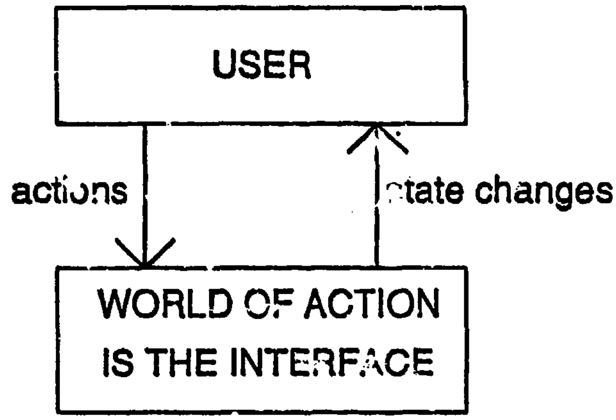

FIGURE 5. The Model-World Interface. Here the user takes action directly in the world of action which is itself the medium for the interface language. The user directly observes state changes in the world.

Expressions With Causal Force

In a system built on the model-world metaphor, the interface language itself can be seen as a world where the user can act, a world that changes state in response to user actions. The world of interest is explicitly represented and there is no intermediary between user and world. The world of interest is constructed and manipulated by expressions in the interface language *where those expressions have the character of actions taken in the world of interest.* This collapse of description to action closes the reference gap between the expression and what it represents. The expression becomes what it represents. Giving expressions causal force in the world of interest is the first half of the solution. This is the basis of the magic in the declaration metaphor.

Note that giving the world of interest explicit representation is not by itself sufficient to create a model-world. SHRDLU (Winograd, 1972) and "Put That There" (Bolt, 1980) are two very impressive systems that have continuous representation of the world of interest. Yet neither is a model-world since both are explicitly conversational in nature. The expressions generated by the user are descriptions to be interpreted by an intermediary. In fact both systems were designed as attempts to fill the conversational metaphor's design vacuum. These systems represent an advance over earlier conversational interfaces because they permit a different sort of reference than is possible in conversational settings where the world described by the expressions in the conversation are not present. "Put That There" is especially interesting because it demonstrates the integration of gesture into conversation. Still, the gestures are not actions in the world of interest, but are instead descriptors to be interpreted by the intermediary agent.

Constraining the Generation of Expressions

Although we mostly seem to overlook it, the physical world has a wonderful property. In the physical world, one cannot do that which cannot be done. When we consider declarations in a computer interface language as analogous to actions in the physical world, the beauty of this property becomes apparent. The constraints of the world are manifest in our interaction with the world. This is just the property we need to prevent the bumbling user/magician from composing an impossible expression. Thus, one solution to the problem of the generation of inappropriate expressions is to build the constraints of the world referred to into the tools the user has for constructing expressions about that world. I have in mind a special sense of building the constraints of the domain into the interface language. I do not mean to make the constraints of the domain syntactic constraints in the language. Many

programming languages attempt to do this by building the logic of the programming world into the syntax of the language. For example, strict typing and type checking in some programming languages makes it a syntax error to do a floating point division on an integer. All this has done is to make it a syntax violation to describe in the interface language that which is not possible in the domain of action. That is just what we don't want. What we do want is to make it impossible to even generate a description of that which is not possible in the domain of action. If we do that, we can collapse the reference relation between description and action into one of identity. Generating the description *is* doing the action.

Consider a simple truly constraining situation. If one has a keyboard with a certain set of characters, then one is constrainted to type only those characters that are available. This is the *only* built-in constraint that exists on most "conversational" interfaces. Additional constraints might be built-in by following the model of operational interlocks on certain devices. Microwave ovens, for example, are designed with interlocks that prevent starting the oven with the door open. In a similar way, one might imagine building a constraint for English text entry that only permitted the letter *u* to be typed after *q*. Once having typed *q*, the only key that would generate a character would be *u*. All others would signal an error. (As silly as it seems, this is not far removed from the nature of many interfaces.) A better way to enforce this constraint might be to only provide the *qu* combination as a pair on a single key. This is the sense in which I intend the "building-in of constraint." It does not mean that the user is enjoined from taking the action, or that an error will be detected and signaled if the user takes that action. It means instead that it is simply not possible, using the tools that the interface language provides, to generate an expression that cannot be realized in the world of action to which the expressions refer.

Such constraints must be embodied in a great deal of structure, and making that structure interpretable requires a good domain metaphor. At present the most obvious way to accomplish this is to build the interface language as a model of a physical world. Perhaps there is some small set of fundamental constraints that must be met in order to support the model-world conception. Something like the existence of objects, that objects do not change unless they are acted upon, that actions may be applied to objects that exist but cannot be applied to objects that do not exist, that objects that exist may be seen, that objects that do not exist cannot be seen, and so on.[4] These constraints on the generation of input expressions are the basis for the claims by proponents of "direct manipulation" that error messages are not required in these systems. The key here again is in the reference relations between the language and the things referred to. The constraints are built into the model world, which serves a dual function as the world of interest and as the medium for the language of interaction. This is the other half of the solution, constraining the magician's language so that only meaningful spells can be uttered. This is what keeps the magic from breaking, what prevents the model-world metaphor from falling apart.

The structure that is present in the interface must be recognizable by the user. There must be a coherent scheme for the operation of the model-world, one that makes sense so that the limitations on the formation of expressions is unnoticed. This is the role of the domain metaphors. Choosing an appropriate domain metaphor that will support the importation of useful structure to the task at hand is critical to the ease of use of such systems. Different domain metaphors have different structures that have different computational properties. Each way of conceiving of a problem may make some things easy to see and other things difficult to see. While the model-world metaphor eliminates referential distance, semantic distance remains an issue. The design of a task domain metaphor that efficiently captures users' intentions is an important component of a usable model-world interface.

Of course, it is always possible to view an interface language that supports the model-world metaphor as a medium for the communication between a user and an intermediary. While both interpretations are available, the choice between them makes a difference. In particular, there is a different sort of relationship between expressions in the input language and the things they refer to in the two cases.

[4] Of course, model worlds need not simulate the properties of the physical world. One of the virtues of the plasticity of the computer medium is that worlds can exist there that could not have a physical reality.

Under the conversational metaphor, the reference relation is as it is in natural language. An expression in the interface language is a symbolic description that refers to actions and objects. Input expressions are interpreted by the intermediary and the actions are carried out upon the objects. Output expressions are interpreted by the user as descriptions of the system state. In the model-world metaphor, both input and output expressions appear to *be* what they refer to. Expressing the action and doing the action are experienced as the same thing.

For example, consider moving an icon in a graphical editor. The movements of the mouse and the clicks required to pick up the icon and put it down somewhere else constitute a complex expression in the interface language. The system designer can see this as an expression in the interface language. But for the user, the editor presents a graphical world in which those actions that comprise the expression in the interface language *are* the actions to be taken on the icon object. The graphical representation of the icon is an expression in the interface output language, and it is also the object being manipulated.

Given this analysis of the components of the model-world metaphor, let us return to Schneiderman's criteria for "direct manipulation" systems. These can now be seen as descriptions of features that help support the model-world metaphor. His requirements of continuous representation of the objects of interest and immediate response are elements that support the creation of the world itself. Schneiderman's notion that one should interact with the system via "physical actions or labeled button presses instead of complex syntax" seems a bit confused, but is clearly on the right track. The heart of the matter is that the expressions in the interface language (however they may be manifested) must be actions in the world of interest itself. Schniederman's call for the reversability of actions is not an inherent property of model worlds in general. Whether it should or should not be a property of the domain metaphor for the model world depends upon the task. In order to support the model-world metaphor, the world must be continuously represented and the consequences of the actions must be as nearly immediate as is possible. But it is not just these features, it is the reference relations that are critical. What is "direct" about direct manipulation is the collapse of description into action, the elimination of the reference gap between the expressions in the interface language and their referents. When we make the interface language the world of interest, we do two things. First, we make expressions into actions. This collapses the reference gap and banishes the implied intermediary. Second, we make the constraints of the world of interest into the constraints on the production of expressions. This provides a natural way to prevent the user from composing an expression that cannot be realized.

Problems in a Model World

Interfaces built on the model-world metaphor suffer from a number of problems. They have recently become quite popular in the commercial marketplace, but they may not yet have come up against their inherent limitations.

As I have tried to demonstrate, the model world collapses symbolic reference and banishes the intermediary who interprets the expressions in the interface language. Surely, one is giving up something when one walks away from several millenia of progress grounded in symbolic reference. Direct manipulation schemes have always been vulnerable to criticisms that they become cumbersome when applied to tasks that can take advantage of the power of abstract reference. Suppose I want to perform some action on every word in this paper that begins with the letter *s*? If I had an agent that understood symbolic descriptions, I could ask it to find all such instances and perform the desired action without knowing in advance how many or where they were. If I were dealing with a model world, what could I do? Would I have to find every instance and act upon it in person, as it were? One way around this problem is to acknowledge that the description specification task and the task that operates on instances are at different levels of user intention. One could imagine then a model world that contains as its objects elements of descriptions and operations. The user could then operate directly in that world to compose the desired abstract action specification to be mapped across the instances in the world where action is ultimately desired (the text file, for example). This is a solution that preserves the model-

world metaphor at a superficial level, in as much as the user directly constructs the abstract description, but what shall we say of the subsequent application of that description? Is that not the action of a new intermediary that the user has brought into existence via action in the model world? This is a difficult question, and I have no easy answers.

The facts that the conversation and model-world metaphors seem to be capable of fading into each other in spite of their fundamental differences and that they can be combined as in collaborative manipulation interfaces raise the question of the importance of maintaining a consistent metaphor throughout an interaction. I take this to be essentially an empirical question, for which there is as yet no answer that I know of. However, it seems quite reasonable to assume that these metaphors are something like points of view on the interface, and there is empirical evidence, in the realm of text comprehension at least, that changes in point of view can interfere with the comprehension of text (Abelson, 1975; Black, Turner, & Bower, 1979).

Finally, the dictum that model worlds shall provide continuous representation of the objects of interest is very difficult to satisfy in worlds of even moderate complexity. Screen real estate is quickly exhausted. And if everything of interest cannot be legibly presented at one time, then measures will have to be taken to provide for display control.

THE COLLABORATIVE MANIPULATION METAPHOR

All of these metaphors are inspired by ideas about the nature of human action and interaction in the absence of computers. The conversation metaphor is based on the assumption that the computer should be an actor in the setting in which it works, and that in order to make it easy for humans to deal with it, it should behave as a human does in human-human interaction. A conversation or dialogue is taken to be the prototypic human-human interaction mode, so the computer is designed to support a conversational interaction. The model-world metaphor rests on the assumption that one of the things that people are really good at is manipulating objects in their environment. The activities of a craftsman may be taken as the prototype for the development of such interfaces. The fact that there are settings in which conversation coexists with the manipulation of objects in the world suggests that these two metaphors might be productively combined in the design of computer interfaces.

For the past several years I have been studying navigation on large ships. In particular I have been looking at the activities of a team of from four to six people who keep track of a ship's position while it is entering or leaving a narrow and congested harbor (San Diego). In this world of navigation, there are many structured representational media that are manipulated by the people in the course of doing the task. These include the navigation chart, plotting tools, measurement tools, written records, reference tables, etc. This is a highly evolved (in the cultural sense) activity and some of the representational media have beautiful computational properties. For example, in plotting a position, a representational state is imposed on a plotting device, and that device is then brought into coordination with the structure of the nautical chart by superimposing it upon the chart. Because of the structure of these representational media, a complex computation can be realized via a few simple alignment procedures. But the fact that this simple superimposition of structure does get the right answer depends critically upon the properties of the plotting tool and the chart itself, which are artifacts that have been created by people who are not present at the occasion of their use.

Consider the relationship between the cartographer who created the chart and the navigator who uses it as one kind of "collaborative manipulation." Every time someone plots a position on the chart, it is a collaboration with the cartographer. Even though the full computation is distributed across space and time and social organization, it is only accomplished by the cartographer and the navigator collaboratively manipulating the computational artifacts of this world. The cartographer could not anticipate where on the chart a ship might be, but had strong expectations about the nature of the procedures that would be used to plot the position and constructed the chart in such a way that those procedures would in fact work.

There is a more immediate sense of collaborative manipulation in the concurrent joint activities of the members of the navigation team. While there is a nominal division of labor among the team members, several of them are co-located in a shared space with shared access to several of the representational technologies. In the process of computing the ship's location, they collaborate in the manipulation of the representational artifacts. Two people may work together to align a plotting tool for a line of position on the chart, or one person may anticipate the needs of another and manipulate a medium to put it in a state from which the other can proceed more easily. Sometimes they achieve coordination with each other by manipulating the structure of the representational artifacts in their environment; sometimes they manipulate the structure of sound waves in the air in their environment; sometimes they gesture and touch each other.

Here we have two instances of "collaborative manipulation" in a real-world task setting. How might they be mapped into the design of a computer interface? Well, consider the situation of any of the people in the navigation setting. The environment contains artifacts and other humans. This person converses with the other people, and manipulates the objects in the environment. But the other people are manipulating those objects as well, and sometimes the communication among the people is conducted via the manipulation of those objects. This suggests a system that contains both a model-world and an intelligent agent. The user should be able to have a conversation about the world with the agent, and both the user and the agent should be able to manipulate the shared world. Figure 6 shows the relation of user to agent and world of action under the collaborative manipulation metaphor.

Command Completion

As a very simple example, consider command completion, a feature that has been around for a long time in some systems. A command language interface is "conversational" in the sense that the user provides descriptions of actions to be taken by an intermediary in some world. At the level of task performance, therefore, the interface is not a model world. At the level of the specification of the character strings that constitute the commands, however, it is usually experienced as a model world. The user takes actions (presses keys) and sees the consequences immediately. Command completion facilities are a way for the interface itself to anticipate, on the basis of partial input, what the user intends, and to use that anticipation to collaboratively manipulate the world that the user is manipulating. Typically, the user types a few characters of a command, then types <space> to signal the collaborator that it

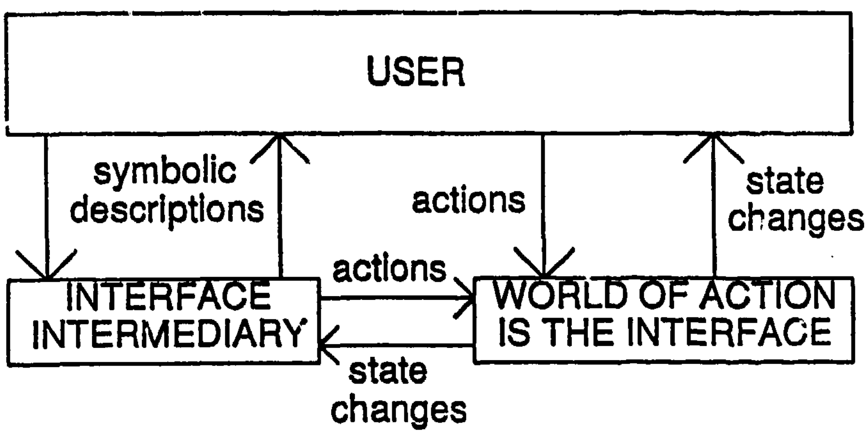

FIGURE 6. The Collaborative Manipulation Interface. This is a combination of the conversation and model-world interfaces. Here the user may interact with an intermediary that can act upon the the world of action, or the user may act upon that world directly.

should attempt to type the remaining characters.[5] We can consider this activity from the point of view of each of the metaphors. Seen via the conversational metaphor, the computer completes one's utterance, just as a good conversational partner might do. From the model-world perspective, the user and machine are engaged in a collaborative manipulation of the user's input. But notice how this last point reflects back onto the human-human conversational setting. An important aspect of conversation is that it is collaborative manipulation of the expressions in the speech channel. When the type of "doing" we are concerned with is "saying," then "saying is doing." It sounds silly, but it is simply another instance of the collapse of the reference gap. In the same way that a conversational interface normally gives a direct manipulation interface to the task of producing character strings, so speech gives us a direct manipulation interface to the production of phonetic sequences.

Two extentions to the Steamer system have been built on a collaborative manipulation metaphor.[6] One is an intelligent display controller for process monitoring situations (McCandless, 1986), and the other is an intelligent knowledge-based graphic designer's aid (Weitzman, 1986). I discuss these below.

Display controller. Displays of the type that can be easily created in Steamer can be connected to real-time processes as well as to simulation models. In typical applications users can choose which display they would like to attend to at any point in time. One of the problems in real-time process monitoring is that the operator forms hypotheses about the state of the process and may subsequently search for information that confirms the hypothesis while disregarding evidence that conflicts with the hypothesis. One way to solve this problem is to have another mind present with other hypotheses. Such a "doubting Thomas" may point to other information. Our group at UCSD has implemented an intelligent display controller that selects displays and display components based upon the "importance" of the process variables that are indicated by the display components. The process variables themselves know when they are in or out of their normal operating ranges, for example, and the display controller can give priority to display components that report the values of variables that are out of range. In fact, the controller is implemented as a parallel distributed processing network that is capable of learning trends in values that precede "important" events, so it can anticipate states of the process and can give variables that are moving in a direction that is ominous in the current context display priority before their values actually become alarming. When the display controller presents the operator with a display, the operator may reject display components, indicating that they are not relevant in the current context.[7] The display controller then learns about the operator's preferences in the same way it learned about the system's behavior: by observation. In this system, the display is the shared world of action. The contents of the display are collaboratively manipulated by the operator and the display controller.

Graphic design aid. The graphics editor that was developed in connection with the Steamer project permits subject matter experts with no computing expertise to generate diagrams (which are actually complex lisp programs) simply by assembling them in a model-world environment. These subject matter experts are seldom expert graphic designers, so the diagrams they create, while capturing something of the subject matter expert's expertise, may be of poor graphic design quality and may not be stylistically similar to each other. *Designer* is an expert system that shares the diagram with the user as a model world for action. The user can have the designer system analyze the diagram. *Designer* will find violations of design principles and notify the user. Furthermore, the user can ask the system to demonstrate ways to correct the violations. Demonstration is an important interface event because it

[5] Although it seems to share some features with command completion, the Do What I Mean (DWIM) facility (Teitelman, 1974) in Interlisp does not belong here. DWIM frequently simply makes the most likely interpretation of the user's input and executes that without notifying the user that it is doing so. DWIM is an intelligent agent, but the input expression itself is never object of discussion, so there is no shared world of action.

[6] At the time these systems were designed, collaborative manipulation was not part of our vocabulary in the laboratory, but the ideas that term refers to were clearly present.

[7] This does admit the possibility of the operator perseverating on a faulty interpretation.

implies collaborative manipulation. The agent performing the demonstration must have direct access to the world, and the actions performed in the demonstration are the content of the communication to the other agent. The ability of the agent who receives the demonstration to act in that world is a presupposition of the demonstration act.

DISCUSSION

Looking back across the several metaphors, we can see relationships between the nature of the technology available and the metaphors for interaction. In the case of the teletype, we could see that technology can suggest metaphors, or at least constrain the sorts of mode of interaction metaphors that are supportable. Teletype technology supports a conversation, figuratively speaking, between user and machine, while high-resolution bitmaps and point devices suggest model worlds. But the metaphor can also constrain the possibilities we see in the technology. The conversation metaphor, in its narrow sense, steers us away from the declaration metaphor by emphasizing the presence of an intermediary. The declaration metaphor is an example of a change in the power of the interface that is brought about not by a change in technology, but by a change in interface metaphor. When users discover the declaration metaphor, they are discovering a mode of interaction that is possible in the technology of the interface but which is not seen under the conversation metaphor.

The choice of a mode of interaction metaphor can make great differences in the power of an interface. We are often not aware of having chosen a particular metaphor, and do not often consider the options available and their computational properties. In this paper I have argued for the viability of two metaphors in addition to the conversational metaphor: the model world and a hybrid, collaborative manipulation. The key to the properties of the interface lies in the reference relations between the expressions in the interface language and the things to which they refer. There are advantages in the abstractness and the ambiguity of symbolic descriptions. There are also gains to be had in taking advantage of the magical character of the worlds that exist on computers. They can be designed in such a way that "saying is doing," and this can be exploited to give the user great ease of interaction. Supporting that ease of interaction, however, leads to limitations on the language that may prevent it reaching the power of the symbolic description mode of interaction.

The issue is clearly not a question of which metaphor is the "best." I only hope we can recognize that metaphors are present at all stages of interface design and use and that they have important consequences. I also hope we can realize that we have, in some sense, been captured by one of several possible metaphors. My reasons for hoping we can come to this vision are, in fact, my reservations about the assumptions underlying the conversation metaphor. First, taking the problem of human-computer interaction to be a communicational problem assumes that the computer will be another intelligent agent rather than a tool or a structured medium that the user can manipulate. It may be that computers will have an important role as agents, but it is certain that they will be a vital class of tool. Communication should not be the only organizing metaphor for human-computer interaction. Second, assuming that human-human communication is acheived primarily via conversation removed from the objects referred to may be a mistake. In face-to-face conversation, a world is present that may contain objects or events to which the conversation refers. This makes reference different in that one can refer to a seen world, and it means that other modes of communication beside speech are available, e.g., demonstration. Looking at the interactions of individuals in a highly evolved real-world task setting we see conversation, but we also see the collaborative manipulation of representational media. Conversation is good when the nature of the task needs to be negotiated or the division of labor is not specified, but when the task is well understood, little conversation needs to take place. In highly evolved task settings, a good deal of the expertise of the system as a whole is in the structure of the artifacts rather than in the people themselves. Third, the skills that people have dealing with each other are adaptations to the limitations of people. It may be that a computer could be even easier for a person to deal with than another person would be. Seeking to imitate human behavior with computers that are to have

roles in task performances may be setting the wrong sort of standard of performance. This criticism applies to all of the interface metaphors discussed in this paper, since all of them are based on mappings from interaction with noncomputational systems. Because computers can manifest behaviors that are not possible in any other medium, we should use our imaginations in the design process. Perhaps as technology develops, we will be able to think of the human as the limited partner in the interaction and design, not another human, but an environment that complements the abilities of human users.

I take these caveats as reminders that the space of interfaces is larger than we have assumed and that it may be larger than we can presently imagine. Given the power of metaphors to change the phenomenological feel of interfaces and the influence of model of interaction metaphors on the direction of development of technology, we, as designers, have a responsibility to give careful consideration to the metaphors we use.

REFERENCES

Abelson, R. P. (1975). Does a story understander need a point of view? In R. Schank & B. Nash-Webber (Eds.), *Theoretical issues in natural language processing*. Washington, DC: Association for Computational Linguistics.

Black, J., Turner, T., & Bower, G. (1979). Point of view in narrative comprehension, memory, and production. *Journal of Verbal Learining and Verbal Behavior, 13*, 559-572.

Bolt, R. (1980). "Put That There": Voice and gesture at the graphics interface. In J. Thomas (Ed.), *Computer graphics: Siggraph '80 conference proceedings* (pp. 262-270). New York: ACM.

D'Andrade, R. (1981). The cultural part of cognition. *Cognitive Science, 5*, 179-195.

Hollan, J. D., Hutchins, E., McCandless, T., Rosenstein, M., & Weitzman, L. (in press). Graphical interfaces for simulation. In W. Rouse (Ed.), *Advances in man-machine systems research* (Vol. 3). Greenwich, CT: JAI Press.

Hollan, J., Hutchins, E., & Weitzman, L. (1984, Summer). Steamer: An interactive inspectable simulation-based training system. *AI Magazine*, pp. 15-27.

Hutchins, E., Hollan, J. D., & Norman, D. A. (1985). Direct manipulation interfaces. *Human-Computer Interaction, 1*, 311-338.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. Chicago: University of Chicago Press.

McCandless, T. (1986). PDP mechanisms for intelligent display control. In P. Luker & H. Adelsberger (Eds.), *Intelligent simulation environments: Proceedings of the Conference on Intelligent Simulation Environments*. Society for Computer Simulation.

Owen, D. (1986). *Interface objects and levels*. Unpublished manuscript. University of California, San Diego, Institute for Cognitive Science, Intelligent Systems Group, La Jolla, CA.

Schneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology, 1*, 237-256.

Schneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer, 16*(8), 57-69.

Searle, J. (1979). *Expression and meaning: Studies in the theory of speech acts.* New York: Cambridge University Press.

Sutherland, I. (1963). Sketchpad: A man-machine graphical communication system. *Proceedings of the Spring Joint Computer Conference* (pp. 329-346). Baltimore, MD: Spartan Books.

Teitelman, W. (1974). *Interlisp Reference Manual.* Palo Alto, CA: Xerox Palo Alto Research Center.

Weitzman, L. (1986). *Designer: A knowledge-based graphic design assistant* (ICS Rep. No. 8609). La Jolla: University of California, San Diego, Institute for Cognitive Science.

Winograd, T. (1972). *Understanding natural language.* New York: Academic Press.

ICS Technical Report List

The following is a list of publications by people in the Institute for Cognitive Science. For reprints, write or call:

Institute for Cognitive Science, C-015
University of California, San Diego
La Jolla, CA 92093
(619) 534-6771

8301. David Zipser. *The Representation of Location*. May 1983.

8302. Jeffrey Elman and Jay McClelland. *Speech Perception as a Cognitive Process: The Interactive Activation Model*. April 1983. Also published in N. Lass (Ed.), *Speech and language: Volume 10*, New York: Academic Press, 1983.

8303. Ron Williams. *Unit Activation Rules for Cognitive Networks*. November 1983.

8304. David Zipser. *The Representation of Maps*. November 1983.

8305. The HMI Project. *User Centered System Design: Part I, Papers for the CHI '83 Conference on Human Factors in Computer Systems*. November 1983. Also published in A. Janda (Ed.), *Proceedings of the CHI '83 Conference on Human Factors in Computing Systems*. New York: ACM, 1983.

8306. Paul Smolensky. *Harmony Theory: A Mathematical Framework for Stochastic Parallel Processing*. December 1983. Also published in *Proceedings of the National Conference on Artificial Intelligence, AAAI-83*, Washington DC, 1983.

8401. Stephen W. Draper and Donald A. Norman. *Software Engineering for User Interfaces*. January 1984. Also published in *Proceedings of the Seventh International Conference on Software Engineering*, Orlando, FL, 1984.

8402. The UCSD HMI Project. *User Centered System Design: Part II, Collected Papers*. March 1984. Also published individually as follows: Norman, D.A. (1984), Stages and levels in human-machine interaction, *International Journal of Man-Machine Studies, 21*, 365-375; Draper, S.W., The nature of expertise in UNIX; Owen, D., Users in the real world; O'Malley, C., Draper, S.W., & Riley, M., Constructive interaction: A method for studying user-computer-user interaction; Smolensky, P., Monty, M.L., & Conway, E., Formalizing task descriptions for command specification and documentation; Bannon, L.J., & O'Malley, C., Problems in evaluation of human-computer interfaces: A case study; Riley, M., & O'Malley, C., Planning nets: A framework for analyzing user-computer interactions; all published in B. Shackel (Ed.), *INTERACT '84, First Conference on Human-Computer Interaction*, Amsterdam: North-Holland,

1984; Norman, D.A., & Draper, S.W., Software engineering for user interfaces, *Proceedings of the Seventh International Conference on Software Engineering*, Orlando, FL, 1984.

8403. Steven L. Greenspan and Eric M. Segal. *Reference Comprehension: A Topic-Comment Analysis of Sentence-Picture Verification*. April 1984. Also published in *Cognitive Psychology, 16*, 556-606, 1984.

8404. Paul Smolensky and Mary S. Riley. *Harmony Theory: Problem Solving, Parallel Cognitive Models, and Thermal Physics*. April 1984. The first two papers are published in *Proceedings of the Sixth Annual Meeting of the Cognitive Science Society*, Boulder, CO, 1984.

8405. David Zipser. *A Computational Model of Hippocampus Place-Fields*. April 1984.

8406. Michael C. Mozer. *Inductive Information Retrieval Using Parallel Distributed Computation*. May 1984.

8407. David E. Rumelhart and David Zipser. *Feature Discovery by Competitive Learning*. July 1984. Also published in *Cognitive Science, 9*, 75-112, 1985.

8408. David Zipser. *A Theoretical Model of Hippocampal Learning During Classical Conditioning*. December 1984.

8501. Ronald J. Williams. *Feature Discovery Through Error-Correction Learning*. May 1985.

8502. Ronald J. Williams. *Inference of Spatial Relations by Self-Organizing Networks*. May 1985.

8503. Edwin L. Hutchins, James D. Hollan, and Donald A. Norman. *Direct Manipulation Interfaces*. May 1985. Also published in D. A. Norman & S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*, 1986, Hillsdale, NJ: Erlbaum.

8504. Mary S. Riley. *User Understanding*. May 1985. Also published in D. A. Norman & S. W. Draper (Eds.), *User Centered System Design: New Perspectives on Human-Computer Interaction*, 1986, Hillsdale, NJ: Erlbaum.

8505. Liam J. Bannon. *Extending the Design Boundaries of Human-Computer Interaction*. May 1985.

8506. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Internal Representations by Error Propagation*. September 1985. Also published in D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Vol. 1. Foundations*, 1986, Cambridge, MA: Bradford Books/MIT Press.

8507. David E. Rumelhart and James L. McClelland. *On Learning the Past Tense of English Verbs*. October 1985. Also published in J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Vol. 2. Psychological and Biological Models*, 1986, Cambridge, MA: MIT Press/Bradford Books.

8601. David Navon and Jeff Miller. *The Role of Outcome Conflict in Dual-Task Interference*. January 1986.

8602. David E. Rumelhart and James L. McClelland. *PDP Models and General Issues in Cognitive Science*. April 1986. Also published in D. E. Rumelhart, J. L. McClelland, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*, 1986, Cambridge, MA: MIT Press/Bradford Books.

8603. James D. Hollan, Edwin L. Hutchins, Timothy P. McCandless, Mark Rosenstein, and Louis Weitzman. *Graphical Interfaces for Simulation*. May 1986. To be published in W. B. Rouse (Ed.), *Advances in Man-Machine Systems* (Vol. 3). Greenwich, CT: Jai Press.

8604. Michael I. Jordan. *Serial Order: A Parallel Distributed Processing Approach*. May 1986.

8605. Ronald J. Williams. *Reinforcement Learning in Connectionist Networks: A Mathematical Analysis*. June 1986.

8606. David Navon. *Visibility or Disability: Notes on Attention*. June 1986.

8607. William Appelbe, Donald Coleman, Allyn Fratkin, James Hutchison, and Walter J. Savitch. *Porting UNIX to a Network of Diskless Micros*. June 1986.

8608. David Zipser. *Programming Neural Nets to Do Spatial Computations*. June 1986. To be published in N. E. Sharkey (Ed.), *Advances in Cognitive Science* (Vol. 2). Norwood, NJ: Ablex.

8609. Louis Weitzman. *Designer: A Knowledge-Based Graphic Design Assistant*. July 1986.

8610. Michael C. Mozer. *RAMBOT: A Connectionist Expert System That Learns by Example*. August 1986.

8611. Michael C. Mozer. *Early Parallel Processing in Reading: A Connectionist Approach*. December 1986. To be published in M. Coltheart (Ed.), *Attention and Performance XII*. Hillsdale, NJ: Lawrence Erlbaum Associates.

8701. Jeffrey L. Elman and David Zipser. *Learning the Hidden Structure of Speech*. February 1987.

8702. Garrison W. Cottrell, Paul Munro, and David Zipser. *Image Compression by Back Propagation: An Example of Extensional Programming*. February 1987.

8703. Edwin Hutchins. *Metaphors for Interface Design*. April 1987. Paper presented at NATO-sponsored Workshop on Multimodal Dialogues Including Voice, Venaco, Corsica, France, September 1986.

Earlier Reports by People in the Cognitive Science Lab

The following is a list of publications by people in the Cognitive Science Lab and the Institute for Cognitive Science. For reprints, write or call:

Institute for Cognitive Science, C-015
University of California, San Diego
La Jolla, CA 92093
(619) 452-6771

ONR-8001. Donald R. Gentner, Jonathan Grudin, and Eileen Conway. *Finger Movements in Transcription Typing*. May 1980.

ONR-8002. James L. McClelland and David E. Rumelhart. *An Interactive Activation Model of the Effect of Context in Perception: Part I*. May 1980. Also published in *Psychological Review, 88,5*, pp. 375-401, 1981.

ONR-8003. David E. Rumelhart and James L. McClelland. *An Interactive Activation Model of the Effect of Context in Perception: Part II*. July 1980. Also published in *Psychological Review, 89, 1*, pp. 60-94, 1982.

ONR-8004. Donald A. Norman. *Errors in Human Performance*. August 1980.

ONR-8005. David E. Rumelhart and Donald A. Norman. *Analogical Processes in Learning*. September 1980. Also published in J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum, 1981.

ONR-8006. Donald A. Norman and Tim Shallice. *Attention to Action: Willed and Automatic Control of Behavior*. December 1980.

ONR-8101. David E. Rumelhart. *Understanding Understanding*. January 1981.

ONR-8102. David E. Rumelhart and Donald A. Norman. *Simulating a Skilled Typist: A Study of Skilled Cognitive-Motor Performance*. May 1981. Also published in *Cognitive Science, 6*, pp. 1-36, 1982.

ONR-8103. Donald R. Gentner. *Skilled Finger Movements in Typing*. July 1981.

ONR-8104. Michael I. Jordan. *The Timing of Endpoints in Movement*. November 1981.

ONR-8105. Gary Perlman. *Two Papers in Cognitive Engineering: The Design of an Interface to a Programming System and MENUNIX: A Menu-Based Interface to UNIX (User Manual)*. November 1981. Also published in *Proceedings of the 1982 USENIX Conference*, San Diego, CA, 1982.

ONR-8106. Donald A. Norman and Diane Fisher. *Why Alphabetic Keyboards Are Not Easy to Use: Keyboard Layout Doesn't Much Matter*. November 1981. Also published in *Human Factors, 24*, pp. 509-515, 1982.

ONR-8107. Donald R. Gentner. *Evidence Against a Central Control Model of Timing in Typing*. December 1981. Also published in *Journal of Experimental Psychology: Human Perception and Performance, 8*, pp. 793-810, 1982.

ONR-8201. Jonathan T. Grudin and Serge Larochelle. *Digraph Frequency Effects in Skilled Typing*. February 1982.

ONR-8202. Jonathan T. Grudin. *Central Control of Timing in Skilled Typing*. February 1982.

ONR-8203. Amy Geoffroy and Donald A. Norman. *Ease of Tapping the Fingers in a Sequence Depends on the Mental Encoding*. March 1982.

ONR-8204. LNR Research Group. *Studies of Typing from the LNR Research Group: The role of context, differences in skill level, errors, hand movements, and a computer simulation*. May 1982. Also published in W. E. Cooper (Ed.), *Cognitive aspects of skilled typewriting*. New York: Springer-Verlag, 1983.

ONR-8205. Donald A. Norman. *Five Papers on Human-Machine Interaction*. May 1982. Also published individually as follows: Some observations on mental models, in D. Gentner and A. Stevens (Eds.), *Mental models*, Hillsdale, NJ: Erlbaum, 1983; A psychologist views human processing: Human errors and other phenomena suggest processing mechanisms, in *Proceedings of the International Joint Conference on Artificial Intelligence*, Vancouver, 1981; Steps toward a cognitive engineering: Design rules based on analyses of human error, in *Proceedings of the Conference on Human Factors in Computer Systems*, Gaithersburg, MD, 1982; The trouble with UNIX, in *Datamation*, *27,12*, November 1981, pp. 139-150; The trouble with networks, in *Datamation*, January 1982, pp. 188-192.

ONR-8206. Naomi Miyake. *Constructive Interaction*. June 1982.

ONR-8207. Donald R. Gentner. *The Development of Typewriting Skill*. September 1982. Also published as Acquisition of typewriting skill, in *Acta Psychologica*, *54*, pp. 233-248, 1983.

ONR-8208. Gary Perlman. *Natural Artificial Languages: Low-Level Processes*. December 1982. Also published in *The International Journal of Man-Machine Studies*, *20*, pp. 373-419, 1984.

ONR-8301. Michael C. Mozer. *Letter Migration in Word Perception*. April 1983. Also published in *Journal of Experimental Psychology: Human Perception and Performance*, *9*, *4*, pp. 531-546, 1983.

ONR-8302. David E. Rumelhart and Donald A. Norman. *Representation in Memory*. June 1983. To appear in R. C. Atkinson, G. Lindzey, & R. D. Luce (Eds.), *Handbook of experimental psychology*. New York: Wiley (in press).

Dr. Phillip L. Ackerman
University of Minnesota
Department of Psychology
Minneapolis, MN 55455

Dr. Beth Adelson
Department of Computer Science
Tufts University
Medford, MA 02155

AFOSR,
Life Sciences Directorate
Bolling Air Force Base
Washington, DC 20332

Dr. Robert Ahlers
Code N711
Human Factors Laboratory
Naval Training Systems Center
Orlando, FL 32813

Dr. Ed Aiken
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Steve Andriole
George Mason University
School of Information
Technology & Engineering
4400 University Drive
Fairfax, VA 22030

Technical Director, ARI
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Albert N. Badre
School of Information and
Computer Science
Georgia Institute of
Technology
Atlanta, GA 30332

Dr. Patricia Baggett
University of Colorado
Department of Psychology
Box 345
Boulder, CO 80309

Dr. Eva L. Baker
UCLA Center for the Study
of Evaluation
145 Moore Hall
University of California
Los Angeles, CA 90024

Dr. L. Bettis
343 Old Annapolis Road
Severna Park, MD 21146

Dr. Gautam Biswas
Department of Computer Science
University of South Carolina
Columbia, SC 29208

Dr. John Black
Teachers College
Columbia University
525 West 121st Street
New York, NY 10027

Dr. Jeff Bonar
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Robert Breaux
Code N-095R
Naval Training Systems Center
Orlando, FL 32813

Commanding Officer
CAPT Lorin W. Brown
NROTC Unit
Illinois Institute of Technology
3300 S. Federal Street
Chicago, IL 60616-3793

Dr. John S. Brown
XEROX Palo Alto Research
Center
3333 Coyote Road
Palo Alto, CA 94304

Dr. John Bruer
The James S. McDonnell
Foundation
University Club Tower, Suite 1610
1034 South Brentwood Blvd.
St. Louis, MO 63117

Dr. Bruce Buchanan
Computer Science Department
Stanford University
Stanford, CA 94305

Maj. Hugh Burns
AFHRL/IDE
Lowry AFB, CO 80230-5000

Dr. Mark Burstein
BBN
10 Moulton Street
Cambridge, MA 02238

Dr. Patricia A. Butler
Education Development Center
55 Chapel Street
Newton MA 01260

Dr. Jaime Carbonell
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. Susan Carey
Harvard Graduate School of
Education
337 Gutman Library
Appian Way
Cambridge, MA 02138

Dr. Pat Carpenter
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. John M. Carroll
IBM Watson Research Center
User Interface Institute
P.O. Box 218
Yorktown Heights, NY 10598

Dr. Fred Chang
Strategic Technology Division
Pacific Bell
2600 Camino Ramon
Rm. 3S-453
San Ramon, CA 94583

Dr. Davida Charney
English Department
Penn State University
University Park, PA 16802

Dr. Eugene Charniak
Brown University
Computer Science Department
Providence, RI 02912

Dr. Michelene Chi
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr. L. J. Chmura
Computer Science and Systems
Code: 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Mr. Raymond E. Christal
AFHRL/MOE
Brooks AFB, TX 78235

Dr. Yee-Yeen Chu
Perceptronics, Inc.
21111 Erwin Street
Woodland Hills, CA 91367-3713

Dr. William Clancey
Stanford University
Knowledge Systems Laboratory
701 Welch Road, Bldg. C
Palo Alto, CA 94304

Assistant Chief of Staff
for Research, Development,
Test and Evaluation
Naval Education and
Training Command (N-5)
NAS Pensacola, FL 32508

Dr. Allan M. Collins
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Stanley Collyer
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Dr. Joel Cooper
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. Albert T. Corbett
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Phil Cunniff
Commanding Officer, Code 7522
Naval Undersea Warfare Engineering
Keyport, WA 98345

Brian Dallman
Technology Applications
Officer
3400 TCHTW/TTGXC
Lowry AFB, CO 80230-5100

Brian Dallman
3400 TTW/TTGXS
Lowry AFB, CO 80230-5000

Dr. Natalie Dehn
Department of Computer and
Information Science
University of Oregon
Eugene, OR 97403

Dr. Gerald F. DeJong
Artificial Intelligence Group
Coordinated Science Laboratory
University of Illinois
Urbana, IL 61801

Dr. Denise Dellarosa
Yale University
Department of Psychology
P.O. Box 11A Yale Station
New Haven, CT 06520-7447

Dr. Andrea di Sessa
University of California
School of Education
Tolman Hall
Berkeley, CA 94720

Dr. Stephanie Doan
Code 6021
Naval Air Development Center
Warminster, PA 18974-5000

Dr. Michael Drillings
Office of Basic Research
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333-5600

Defense Technical
Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
(12 Copies)

Dr. Richard Duran
University of California
Santa Barbara, CA 93106

Dr. Susan Epstein
Hunter College
144 S. Mountain Avenue
Montclair, NJ 07042

ERIC Facility-Acquisitions
4833 Rugby Avenue
Bethesda, MD 20014

Dr. Marshall J. Farr
Farr-Sight Co.
2520 North Vernon Street
Arlington, VA 22207

Dr. Paul Feltovich
Southern Illinois University
School of Medicine
Medical Education Department
P.O. Box 3926
Springfield, IL 62708

Mr. Wallace Feurzeig
Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

Dr. Craig I. Fields
ARPA
1400 Wilson Blvd.
Arlington, VA 22209

Distribution list [UCSD/Norman & Hutchins] Nr 667-541

Dr. Gerhard Fischer
University of Colorado
Department of Computer Science
Boulder, CO 80309

Dr. Linda Flower
Carnegie-Mellon University
Department of English
Pittsburgh, PA 15213

Dr. Kenneth D. Forbus
University of Illinois
Department of Computer Science
1304 West Springfield Avenue
Urbana, IL 61801

Dr. Barbara A. Fox
University of Colorado
Department of Linguistics
Boulder, CO 80309

Dr. Carl H. Frederiksen
McGill University
3700 McTavish Street
Montreal, Quebec H3A 1Y2
CANADA

Dr. John R. Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge, MA 02138

Dr. Michael Friendly
Psychology Department
York University
Toronto ONT
CANADA M3J 1P3

Dr. Michael Genesereth
Stanford University
Computer Science Department
Stanford, CA 94305

Dr. Dedre Gentner
University of Illinois
Department of Psychology
603 E. Daniel St.
Champaign, IL 61820

Dr. Robert Glaser
Learning Research
& Development Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15260

Dr. Joseph Goguen
Computer Science Laboratory
SRI International
333 Ravenswood Avenue
Menlo Park, CA 94025

Dr. Susan Goldman
University of California
Santa Barbara, CA 93106

Dr. Sherrie Gott
AFHRL/MODJ
Brooks AFB, TX 78235

Dr. T. Govindaraj
Georgia Institute of Technology
School of Industrial & Systems
Engineering
Atlanta, GA 30332

Dr. James G. Greeno
University of California
Berkeley, CA 94720

Dr. Dik Gregory
Behavioral Sciences Division
Admiralty Research
Establishment
Teddington
Middlesex, ENGLAND

Dr. Henry M. Halff
Halff Resources, Inc.
4918 33rd Road, North
Arlington, VA 22207

Dr. Bruce Hamill
The Johns Hopkins University
Applied Physics Laboratory
Laurel, MD 20707

Dr. John M. Hammer
Center for Man-Machine
Systems Research
Georgia Institute of Technology
Atlanta, GA 30332

Janice Hart
Office of the Chief
of Naval Operations
OP-11HD
Department of the Navy
Washington, D.C. 20350-2000

Prof. John R. Hayes
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Barbara Hayes-Roth
Department of Computer Science
Stanford University
Stanford, CA 95305

Dr. Frederick Hayes-Roth
Teknowledge
525 University Ave.
Palo Alto, CA 94301

Dr. Joan I. Heller
505 Haddon Road
Oakland, CA 94606

Dr. Shelly Heller
Department of Electrical Engi-
neering & Computer Science
George Washington University
Washington, DC 20052

Dr. James D. Hollan
MCC,
Human Interface Program
3500 West Balcones Center Dr.
Austin, TX 78759

Dr. Melissa Holland
Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Keith Holyoak
University of California,
Los Angeles
Department of Psychology
Los Angeles, CA 90024

Dr. James Howard
Dept. of Psychology
Human Performance Laboratory
Catholic University of
America
Washington, DC 20064

Dr. Ed Hutchins
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Icon Project
Department of Computer Science
Gould-Simpson Science Building
The University of Arizona
Tuscon, AZ 87521

Dr. Dillon Inouye
WICAT Education Institute
Provo, UT 84057

Dr. Alice Isen
Department of Psychology
University of Maryland
Catonsville, MD 21228

Dr. Janet Jackson
Rijksuniversiteit Groningen
Biologisch Centrum, Vleugel D
Kerklaan 30, 9751 NN Haren (Gn.)
NETHERLANDS

Dr. R. J. K. Jacob
Computer Science and Systems
Code: 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Dr. Claude Janvier
Directeur, CIRADE
Universite' du Quebec a Montreal
P.O. Box 8888, St. "A"
Montreal, Quebec H3C 3P8
CANADA

Dr. Robin Jeffries
Hewlett-Packard Laboratories
P.O. Box 10490
Palo Alto, CA 94303-0971

Dr. Marcel Just
Carnegie-Mellon University
Department of Psychology
Schenley Park
Pittsburgh, PA 15213

Dr. Ruth Kanfer
University of Minnesota
Department of Psychology
Elliott Hall
75 E. River Road
Minneapolis, MN 55455

Dr. Wendy Kellogg
IBM T. J. Watson Research Ctr.
P.O. Box 218
Yorktown Heights, NY 10598

Dr. Dennis Kibler
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. David Kieras
University of Michigan
Technical Communication
College of Engineering
1223 E. Engineering Building
Ann Arbor, MI 48109

Dr. Peter Kincaid
Training Analysis
& Evaluation Group
Department of the Navy
Orlando, FL 32813

Dr. Janet L. Kolodner
Georgia Institute of
Technology
School of Information and
Computer Science
Atlanta, GA 30332-0280

Dr. Kenneth Kotovsky
Department of Psychology
Community College of
Allegheny County
800 Allegheny Avenue
Pittsburgh, PA 15233

Dr. David H. Krantz
2 Washington Square Village
Apt. # 15J
New York, NY 10012

Dr. John Laird
EECS
University of Michigan
Ann Arbor, MI 48109-2122

Dr. Pat Langley
University of California
Department of Information
and Computer Science
Irvine, CA 92717

Dr. Jill Larkin
Carnegie-Mellon University
Department of Psychology
Pittsburgh, PA 15213

Dr. R. W. Lawler
ARI 6 S 10
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Paul E. Lehner
PAR Technology Corp.
7926 Jones Branch Drive
Suite 170
McLean, VA 22102

Dr. Alan M. Lesgold
Learning Research and
Development Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Alan Leshner
Deputy Division Director
Behavioral and Neural Sciences
National Science Foundation
1800 G Street
Washington, DC 20550

Dr. Jim Levin
Department of
Educational Psychology
210 Education Building
1310 South Sixth Street
Champaign, IL 61820-6990

Dr. John Levine
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Dr. Clayton Lewis
University of Colorado
Department of Computer Science
Campus Box 430
Boulder, CO 80309

Matt Lewis
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Library,
Naval War College
Newport, RI 02940

Library,
Naval Training Systems
Center
Orlando, FL 32813

Science and Technology Division,
Library of Congress
Washington, DC 20540

Dr. Marcia C. Linn
Lawrence Hall of Science
University of California
Berkeley, CA 94720

Dr. Don Lyon
P. O. Box 44
Higley, AZ 85236

Dr. Stuart Macmillan
FMC Corporation
Central Engineering Labs
1185 Coleman Avenue, Box 580
Santa Clara, CA 95052

Vern Malec
NPRDC, Code P-306
San Diego, CA 92152-6800

Dr. Jane Malin
Mail Code SR 111
NASA Johnson Space Center
Houston, TX 77058

Dr. Sandra P. Marshall
Dept. of Psychology
San Diego State University
San Diego, CA 92182

Dr. Manton M. Matthews
Department of Computer Science
University of South Carolina
Columbia, SC 29208

Dr. Richard E. Mayer
Department of Psychology
University of California
Santa Barbara, CA 93106

Dr. Joe McLachlan
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. James S. McMichael
Navy Personnel Research
and Development Center
Code 05
San Diego, CA 92152

Prof. D. Michie
The Turing Institute
36 North Hanover Street
Glasgow G1 2AD, Scotland
UNITED KINGDOM

Dr. George A. Miller
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. James R. Miller
MCC
9430 Research Blvd.
Echelon Building #1, Suite 231
Austin, TX 78759

Dr. Lance Miller
IBM-FSD Headquarters
6600 Rockledge Drive
Bethesda, MD 20817

Dr. Nancy Morris
Search Technology, Inc.
5550-A Peachtree Parkway
Technology Park/Summit
Norcross, GA 30092

Dr. Randy Mumaw
Program Manager
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. Allen Munro
Behavioral Technology
Laboratories - USC
1845 S. Elena Ave., 4th Floor
Redondo Beach, CA 90277

Dr. T. Niblett
The Turing Institute
36 North Hanover Street
Glasgow G1 2AD, Scotland
UNITED KINGDOM

Dr. Richard E. Nisbett
University of Michigan
Institute for Social Research
Room 5261
Ann Arbor, MI 48109

Dr. A. F. Norcio
Computer Science and Systems
Code: 7590
Information Technology Division
Naval Research Laboratory
Washington, DC 20375

Dr. Donald A. Norman
Institute for Cognitive
Science C-015
University of California, San Diego
La Jolla, California 92093

Deputy Technical Director,
NPRDC Code 01A
San Diego, CA 92152-6800

Director, Training Laboratory,
NPRDC (Code 05)
San Diego, CA 92152-6800

Director, Manpower and Personnel
Laboratory,
NPRDC (Code 06)
San Diego, CA 92152-6800

Director, Human Factors
& Organizational Systems Lab,
NPRDC (Code 07)
San Diego, CA 92152-6800

Library, NPRDC
Code P201L
San Diego, CA 92152-6800

Technical Director,
Navy Personnel R&D Center
San Diego, CA 92152-6800

Commanding Officer,
Naval Research Laboratory
Code 2627
Washington, DC 20390

Dr. Stellan Ohlsson
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Director, Research Programs,
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1133
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142BI
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142
800 N. Quincy St.
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142PS
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 1142CS
800 N. Quincy Street
Arlington, VA 22217-5000
(6 Copies)

Office of Naval Research,
Code 11R
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research,
Code 125
800 N. Quincy Street
Arlington, VA 22217-5000

Psychologist,
Office of Naval Research
Branch Office, London
Box 39
FPO New York, NY 09510

Special Assistant for Marine
Corps Matters,
ONR Code 00MC
800 N. Quincy St.
Arlington, VA 22217-5000

Psychologist,
Office of Naval Research
Liaison Office, Far East
APO San Francisco, CA 96503

Office of Naval Research,
Resident Representative,
UCSD
University of California,
San Diego
La Jolla, CA 92093-0001

Dr. Judith Orasanu
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard St.
Alexandria, VA 22311

Dr. Everett Palmer
Mail Stop 239-3
NASA-Ames Research Center
Moffett Field, CA 94035

Prof. Seymour Papert
20C-109
Massachusetts Institute
of Technology
Cambridge, MA 02139

Dr. Roy Pea
New York University
32 Washington Place
Room 23
New York, NY 10003

Dr. Douglas Pearse
DCIEM
Box 2000
Downsview, Ontario
CANADA

Dr. Nancy Pennington
University of Chicago
Graduate School of Business
1101 E. 58th St.
Chicago, IL 60637

Military Assistant for Training and
Personnel Technology,
OUSD (R & E)
Room 3D129, The Pentagon
Washington, DC 20301-3080

Dr. David N. Perkins
Educational Technology Center
337 Gutman Library
Appian Way
Cambridge, MA 02138

Dr. Steven Pinker
Department of Psychology
E10-018
M.I.T.
Cambridge, MA 02139

Dr. Tjeerd Plomp
Twente University of Technology
Department of Education
P.O. Box 217
7500 AE ENSCHEDE
THE NETHERLANDS

Dr. Martha Polson
Department of Psychology
Campus Box 346
University of Colorado
Boulder, CO 80309

Dr. Peter Polson
University of Colorado
Department of Psychology
Boulder, CO 80309

Dr. Steven E. Poltrock
MCC,
Human Interface Program
3500 West Balcones Center Dr.
Austin, TX 78759

Dr. Harry E. Pople
University of Pittsburgh
Decision Systems Laboratory
1360 Scaife Hall
Pittsburgh, PA 15261

Dr. Joseph Psotka
ATTN: PERI-1C
Army Research Institute
5001 Eisenhower Ave.
Alexandria, VA 22333

Dr. Lynne Reder
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Steve Reder
Northwest Regional Lab.
300 SW Sixth Ave.
Portland, OR 97204

Dr. James A. Reggia
University of Maryland
School of Medicine
Department of Neurology
22 South Greene Street
Baltimore, MD 21201

Dr. Wesley Regian
AFHRL/MOD
Brooks AFB, TX 78235

Dr. Fred Reif
Physics Department
University of California
Berkeley, CA 94720

Dr. Brian Reiser
Department of Psychology
Green Hall
Princeton University
Princeton, NJ 08540

Dr. Lauren Resnick
Learning R & D Center
University of Pittsburgh
3939 O'Hara Street
Pittsburgh, PA 15213

Dr. Gil Ricard
Mail Stop C04-14
Grumman Aerospace Corp.
Bethpage, NY 11714

Dr. J. Jeffrey Richardson
Center for Applied AI
Graduate School of Business
and Administration
University of Colorado
Boulder, CO 80309-0419

Mark Richer
1041 Lake Street
San Francisco, CA 94118

Dr. Edwina L. Rissland
Dept. of Computer and
Information Science
University of Massachusetts
Amherst, MA 01003

Dr. Linda G. Roberts
Science, Education, and
Transportation Program
Office of Technology Assessment
Congress of the United States
Washington, DC 20510

Dr. Ernst Z. Rothkopf
Teachers College
Columbia University
525 West 121st Street
New York, NY 10027

Dr. William B. Rouse
Search Technology, Inc.
5550-A Peachtree Parkway
Technology Park/Summit
Norcross, GA 30092

Dr. Roger Schank
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Alan H. Schoenfeld
University of California
Department of Education
Berkeley, CA 94720

Dr. Janet Schofield
Learning R&D Center
University of Pittsburgh
Pittsburgh, PA 15260

Karen A. Schriver
Department of English
Carnegie-Mellon University
Pittsburgh, PA 15213

Dr. Miriam Schustack
Code 51
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Marc Sebrechts
Department of Psychology
Wesleyan University
Middletown, CT 06475

Dr. Colleen M. Seifert
Intelligent Systems Group
Institute for
Cognitive Science (C-015)
UCSD
La Jolla, CA 92093

Dr. Daniel Sewell
Search Technology, Inc.
5550-A Peachtree Parkway
Technology Park/Summit
Norcross, GA 30092

Dr. Sylvia A. S. Shafto
Department of
Computer Science
Towson State University
Towson, MD 21204

Lawrence E. Sheets
Manager, AI Technology
FMC Northern Ordnance
4800 East River Road
Minneapolis, MN 55421

Dr. Ben Shneiderman
Dept. of Computer Science
University of Maryland
College Park, MD 20742

Dr. Ted Shortliffe
Computer Science Department
Stanford University
Stanford, CA 94305

Dr. Valerie Shute
AFHRL/MOE
Brooks AFB, TX 78235

Dr. Herbert A. Simon
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

LTCOL Robert Simpson
Defense Advanced Research
Projects Administration
1400 Wilson Blvd.
Arlington, VA 22209

Dr. Derek Sleeman
Dept. of Computing Science
King's College
Old Aberdeen
AB9 2UB
UNITED KINGDOM

Dr. Elliot Soloway
Yale University
Computer Science Department
P.O. Box 2158
New Haven, CT 06520

Dr. Richard Sorensen
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Kathryn T. Spoehr
Brown University
Department of Psychology
Providence, RI 02912

Dr. Robert Sternberg
Department of Psychology
Yale University
Box 11A, Yale Station
New Haven, CT 06520

Dr. Kurt Steuck
AFHRL/MOD
Brooks AFB
San Antonio TX 78235

Dr. Albert Stevens
Bolt Beranek & Newman, Inc.
10 Moulton St.
Cambridge, MA 02238

Dr. Paul J. Sticha
Senior Staff Scientist
Training Research Division
HumRRO
1100 S. Washington
Alexandria, VA 22314

Dr. John Tangney
AFOSR/NL
Bolling AFB, DC 20332

Dr. Perry W. Thorndyke
FMC Corporation
Central Engineering Labs
1185 Coleman Avenue, Box 580
Santa Clara, CA 95052

Dr. Martin A. Tolcott
Decision Science
Consortium, Inc.
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Douglas Towne
Behavioral Technology Labs
1845 S. Elena Ave.
Redondo Beach, CA 90277

Headquarters, U. S. Marine Corps
Code MPI-20
Washington, DC 20380

Dr. Kurt Van Lehn
Department of Psychology
Carnegie-Mellon University
Schenley Park
Pittsburgh, PA 15213

Dr. Ralph Wachter
JHU-APL
Johns Hopkins Road
Laurel, MD 20707

Dr. Beth Warren
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Keith T. Wescourt
FMC Corporation
Central Engineering Labs
1185 Coleman Ave., Box 580
Santa Clara, CA 95052

Dr. Douglas Wetzel
Code 12
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Barbara White
Bolt Beranek & Newman, Inc.
10 Moulton Street
Cambridge, MA 02238

Dr. Heather Wild
Naval Air Development
Center
Code 6021
Warminster, PA 18974-5000

Mr. David C. Wilkins
Stanford University
Knowledge Systems Laboratory
701 Welch Road, Bldg. C
Palo Alto, CA 94304

Dr. Michael Williams
IntelliCorp
1975 El Camino Real West
Mountain View, CA 94040-2216

Dr. Robert A. Wisher
U.S. Army Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333

Dr. Martin F. Wiskoff
Navy Personnel R & D Center
San Diego, CA 92152-6800

Dr. Dan Wolz
AFHRL/MOE
Brooks AFB, TX 78235

Dr. Wallace Wulfeck, III
Navy Personnel R&D Center
San Diego, CA 92152-6800

Dr. Joe Yasatuke
AFHRL/LRT
Lowry AFB, CO 80230

Dr. Masoud Yazdani
Dept. of Computer Science
University of Exeter
Exeter EX4 4QL
Devon, ENGLAND

Mr. Carl York
System Development Foundation
181 Lytton Avenue
Suite 210
Palo Alto, CA 94301

Dr. Joseph L. Young
Memory & Cognitive
Processes
National Science Foundation
Washington, DC 20550

Dr. Steven Zornetzer
Office of Naval Research
Code 114
800 N. Quincy St.
Arlington, VA 22217-5000

Dr. Michael J. Zyda
Naval Postgraduate School
Code 52CK
Monterey, CA 93943-5100